



1K AR
GUNNISON, McKAY & HODGSON, L.L.P.

GARDEN WEST OFFICE PLAZA, SUITE 220

1900 GARDEN ROAD

MONTEREY, CALIFORNIA 93940

(831) 655-0880

FACSIMILE (831) 655-0888

June 25, 2008

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR REVISED APPEAL BRIEF

RE: Applicant(s): Eduard K. de Jong
Assignee: Sun Microsystems, Inc.
Title: RENDERING AND ENCRYPTION ENGINE FOR
APPLICATION PROGRAM OBFUSCATION
Serial No.: 10/672,184 Filed: September 25, 2003
Examiner: Ponnoreay Pich Group Art Unit: 2135
Docket No.: SUN040027

Dear Sir:

Transmitted herewith are the following documents in
Response to the Notice of Non-Compliant Appeal Brief dated May
29, 2008 in the above application:

1. Return receipt postcard;
2. This Transmittal Letter (2 pages); and
3. Revised Appeal Brief (40 pages); and

☒ Conditional Petition for Extension of Time: If an
extension of time is required for timely filing of the
enclosed documents after all papers filed with this
transmittal have been considered, Applicant(s) hereby petition
for such an extension of time.

Transmittal Letter
Serial No. 10/672,184
June 25, 2008

☒ The Commissioner is hereby authorized to charge any additional fees required for consideration of the enclosed documents, and to credit any overpayment of fees to Deposit Account No. 50-0553.

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 25, 2008.



Attorney for Applicant(s)

June 25, 2008

Date of Signature

Respectfully submitted,



Forrest Gunnison
Attorney for Applicant(s)
Reg. No. 32,899

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Eduard K. de Jong

Assignee: Sun Microsystems, Inc.

Title: RENDERING AND ENCRYPTION ENGINE FOR APPLICATION
PROGRAM OBFUSCATION

Serial No.: 10/672,184

Filed: September 25, 2003

Examiner: Ponnoreay Pich

Group Art 2135
Unit:

Docket No.: SUN040027

Monterey, CA
June 25, 2008

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REVISED APPEAL BRIEF

Dear Sir:

Pursuant to 37 CFR § 41.37(a) (1), Appellant files this
Appellant's Brief in support of the Notice of Appeal entered by
the USPTO on May 29, 2008.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

REAL PARTY IN INTEREST

The assignee of the above-referenced patent application, Sun Microsystems, Inc., is the real party in interest.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

RELATED APPEALS AND INTERFERENCES

No other appeals or interferences are known to the undersigned Attorney for Appellant, or the Assignee Appellant, which will directly affect, or be directly affected by, or have a bearing on the Board's decision in this pending Appeal.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

STATUS OF CLAIMS

Claims 1 to 20 are pending. Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19 stand rejected in the Final Office Action of November 26, 2007. Claims 5, 10, 15 and 20 stand withdrawn in the Final Office Action of November 26, 2007. The rejections of Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19 are hereby appealed.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

STATUS OF AMENDMENTS

All amendments to the claims presented by Appellant have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

With respect to Claims 1 to 4, 6 to 9, 11 to 14, and 16 to 19, embodiments in accordance with the present invention provide:

[0011] Obfuscation of an application program comprises receiving an obfuscated key decryption program comprising an instruction stream configured to perform a decryption algorithm for a first cryptographic key. The obfuscated key decryption program also has an encrypted second cryptographic key scrambled in its instruction stream. The second cryptographic key is encrypted with the first cryptographic key. The obfuscated key decryption program is executed to decrypt the second cryptographic key. The second cryptographic key is used for decrypting digital content.

Patent Specification, As filed, pg. 6.

A summary is provided below for each independent claim and for each dependent claim argued separately.

CLAIM 1

With respect to Claim 1, a method for application program obfuscation is recited, (See Figs. 37 and 38.), which is performed on a single entity, an application program provider. In this method, the application program provider receives a reference to a decryption algorithm and a first cryptographic key {Paragraph [0125], 3800, Fig. 38}.

Next, the application program provider creates a key decryption program comprising an instruction stream {Paragraph [0125], 3805, Fig. 38; 3720, Fig. 37}. The instruction stream {3725, Fig. 37} is configured to perform the decryption algorithm for the first cryptographic key {Paragraph [0122]}.

The application program provider also applies a cryptographic process {3710, Fig. 37} to a second cryptographic

key {3710, Fig. 37} to create an encrypted second cryptographic key {3755, Fig. 37; 3810, Fig. 38; Paragraphs [0122] and [00125]}. The cryptographic process {3715} receives the first and second cryptographic keys {3705, 3710} as inputs.

Next, the application program provider scrambles {3740 Fig. 37; 3815, Fig. 38} the encrypted second cryptographic key {3755, Fig. 37} into the instruction stream using a code obfuscation method indicated by an obfuscation descriptor {3730, Fig. 37}. The scrambling creates an obfuscated key decryption program {3780, Fig. 37}. The obfuscation descriptor {3730, Fig. 37} is based at least in part on a target ID {3700, Fig. 37}. The target ID specifies a user device for executing an obfuscated application program. {Paragraphs [0122] and [00125]}

Finally, the application program provider sends the obfuscated key decryption program {3820, Fig. 38, Paragraph [0125]}.

CLAIM 6

With respect to Claim 6, a program storage device readable by a machine {Paragraphs [0015], [0033] Fig. 2}, embodies a program of instructions executable by the machine to perform a method for application program obfuscation. The method for application program obfuscation(See Figs. 37 and 38.) is performed on a single entity, an application program provider. In this method, the application program provider receives a reference to a decryption algorithm and a first cryptographic key {Paragraph [0125], 3800, Fig. 38}.

Next, the application program provider creates a key decryption program comprising an instruction stream {Paragraph [0125], 3805, Fig. 38; 3720, Fig. 37}. The instruction stream {3725, Fig. 37} is configured to perform the decryption algorithm for the first cryptographic key {Paragraph [0122]}.

The application program provider also applies a cryptographic process {3710, Fig. 37} to a second cryptographic key {3710, Fig. 37} to create an encrypted second cryptographic key {3755, Fig. 37; 3810, Fig. 38; Paragraphs [0122] and [00125]}. The cryptographic process {3715} receives the first and second cryptographic keys {3705, 3710} as inputs.

Next, the application program provider scrambles {3740 Fig. 37; 3815, Fig. 38} the encrypted second cryptographic key {3755, Fig. 37} into the instruction stream using a code obfuscation method indicated by an obfuscation descriptor {3730, Fig. 37}. The scrambling creates an obfuscated key decryption program {3780, Fig. 37}. The obfuscation descriptor {3730, Fig. 37} is based at least in part on a target ID {3700, Fig. 37}. The target ID specifies a user device for executing an obfuscated application program. {Paragraphs [0122] and [00125]}

Finally, the application program provider sends the obfuscated key decryption program {3820, Fig. 38, Paragraph [0125]}.

CLAIM 11

With respect to Claim 11, an apparatus for application program obfuscation {315, Fig. 3, 415, Fig. 4} includes a memory and a processor. {See Fig. 2} The memory is coupled to said processor. The memory has stored therein computer readable instructions {Paragraph [0033]}. Execution of the computer readable instructions on said processor provides the following means.

Means for receiving, on an application program provider, a reference to a decryption algorithm and a first cryptographic key. {Paragraph [0125], 3800, Fig. 38}

Means for creating a key decryption program comprising an instruction stream {Paragraph [0125], 3805, Fig. 38; 3720, Fig.

37}. The instruction stream {3725, Fig. 37} is configured to perform the decryption algorithm for the first cryptographic key {Paragraph [0122]}.

Means for applying a cryptographic process {3710, Fig. 37} to a second cryptographic key {3710, Fig. 37} to create an encrypted second cryptographic key {3755, Fig. 37; 3810, Fig. 38; Paragraphs [0122] and [00125]}. The cryptographic process {3715} receives the first and second cryptographic keys {3705, 3710} as inputs.

Next, means for scrambling {3740 Fig. 37; 3815, Fig. 38} the encrypted second cryptographic key {3755, Fig. 37} into the instruction stream using a code obfuscation method indicated by an obfuscation descriptor {3730, Fig. 37}. The scrambling creates an obfuscated key decryption program {3780, Fig. 37}. The obfuscation descriptor {3730, Fig. 37} is based at least in part on a target ID {3700, Fig. 37}. The target ID specifies a user device for executing an obfuscated application program. {Paragraphs [0122] and [00125]}

Finally, means for sending sends the obfuscated key decryption program {3820, Fig. 38, Paragraph [0125]}.

CLAIM 16

With respect to Claim 16, an apparatus for application program obfuscation {315, Fig. 3, 415, Fig. 3} include an application program provider. The application program provider in turn includes a memory and a processor. {See Fig. 2} The memory is coupled to said processor. The memory has stored therein computer readable instructions {Paragraph [0033]}. Execution of the computer readable instructions configures the application program provider to:

receive a reference to a decryption algorithm and a first cryptographic key {Paragraph [0125], 3800, Fig. 38}.
create a key decryption program comprising an instruction stream {Paragraph [0125], 3805, Fig. 38; 3720,

Fig. 37}. The instruction stream {3725, Fig. 37} is configured to perform the decryption algorithm for the first cryptographic key {Paragraph [0122]}.

apply a cryptographic process {3710, Fig. 37} to a second cryptographic key {3710, Fig. 37} to create an encrypted second cryptographic key {3755, Fig. 37; 3810, Fig. 38; Paragraphs [0122] and [00125]}. The cryptographic process {3715} receives the first and second cryptographic keys {3705, 3710} as inputs.

scramble {3740 Fig. 37; 3815, Fig. 38} the encrypted second cryptographic key {3755, Fig. 37} into the instruction stream using a code obfuscation method indicated by an obfuscation descriptor {3730, Fig. 37}. The scrambling creates an obfuscated key decryption program {3780, Fig. 37}. The obfuscation descriptor {3730, Fig. 37} is based at least in part on a target ID {3700, Fig. 37}. The target ID specifies a user device for executing an obfuscated application program. {Paragraphs [0122] and [00125]}

send the obfuscated key decryption program {3820, Fig. 38, Paragraph [0125]}.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19 are unpatentable under 35 U.S.C. § 103(a) over U. S. Patent No. 7,170,999, in view of U.S. Patent No. 6,789,177, further in view of PCT Publication No. WO 02/079955, and still further in view of U.S. Patent Application Publication No. US 2002/0120854 A1?

ARGUMENT

Claims 1, 6, 11 and 16 are patentable.

Claims 1-2, 4, 6-7, 9, 11-12, 14, 16-17, and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U. S. Patent No. 7,170,999, hereinafter referred to as Kessler, in view of U.S. Patent No. 6,789,177, hereinafter referred to as Okada, further in view of PCT Publication No. WO 02/079955, herein after referred to as Shen Orr, and still further in view of U.S. Patent Application Publication No. 2002/0120854, herein after referred to as Levine.

With respect to the obviousness rejection of Claims 1, 6, 11 and 16, Applicant respectfully submits that neither the references nor the claims have been considered as a whole as required by the MPEP in an obviousness rejection and that the basis for the combination of references is not well founded.

Using Claim 1 as an example, Claim 1 recites a method that is performed on a single entity, an application program provider, and that receives as inputs, reference to a decryption algorithm and a first cryptographic key. The method also uses a second cryptographic key, a target ID, a cryptographic process and a code obfuscation method. The method recites specific operations that create and then send a specific program, an obfuscated key decryption program.

The obfuscated key decryption program includes an instruction stream configured to perform a decryption algorithm for the first cryptographic key scrambled with an encrypted second cryptographic key. This interpretation follows directly from the claim itself when the claim is considered as a whole and the words in the claim are given their plain meaning. The MPEP requires that "words of the claim must be given their plain meaning unless the plain meaning is inconsistent with the

specification." MPEP § 2111.01 I., 8th Ed. Rev. 6, p 2100-38 (Sept. 2007).

The rejection is required to consider the claims as a whole. MPEP § 2141.02 I., 8th Ed. Rev. 6, p 2100-123 (Sept. 2007). This was not done. Also, the rejection is required to consider the references as a whole. Id. Instead pieces with names similar to elements in the claims were extracted from the references without regard to the teachings of the references taken as a whole.

The rejection based on Kessler has multiple errors.

For example, with respect to the primary reference Kessler, the rejection stated:

As per claim 1, Kessler discloses receiving a reference to a decryption algorithm a first cryptographic key and creating a key decryption program (i.e. proprietary client software having embedded decryption algorithms) comprising an instruction stream, said key decryption program configured to perform said decryption algorithm for first cryptographic key (col 8, lines 50-67 and col 10, lines 58-67). A person skilled the art should understand that all programs start out as source code and when the source code is compiled, the resulting software contains instruction streams. The cited portion of Kessler discloses of a proprietary client software received by the client. The proprietary software contains decryption algorithms used to access the encrypted data via keys SK2 and TK. The fact that the proprietary software exists implies that a reference to a decryption algorithm (i.e. decryption algorithm source code) was received used to create the proprietary software. According to the cited portion in column 8, keys used in the decryption algorithm are obfuscated and/or encrypted in the proprietary software. This implies that when the proprietary software was created, not was the source code to the decryption algorithm received, but also the first decryption key, i.e. SK2, which is used to decrypt TK. (All emphasis in original)

Kessler discloses applying a cryptographic process to a second cryptographic i.e. TK, to create an encrypted second cryptographic key wherein said cryptographic process receives a public key and second cryptographic

keys as inputs result of a track key (TK) being encrypted and decrypted using a single first cryptographic key rather than use of an asymmetric key pair to encrypt and decrypt TK.

Final Office Action, Pgs. 7, 8, November 26, 2007.

Elements are selectively extracted and recombined in a way that changes the principles of operation of Kessler.

The rejection apparently considers the client computer of "User 1" in the peer-to-peer network that provides access to a file as the application program provider of Claim 1. This is supported by the fact that the rejection relies upon the encryption of key TK, which occurs on the client computer of "User 1." (Kessler, Col. 5, line 29 and 35.) The client computer of "User 2" does not provide anything and instead, according to Kessler, is the receiver of the file transferred by "User 1." If the rejection relies upon some other interpretation of the application program provider based on Kessler, such an interpretation directly contradicts the teachings of Kessler and so cannot stand.

Kessler identifies the public and secret keys associated with "User 1" as PK1 and SK1, respectively. (Kessler, Col. 4, line 63, 64.) Kessler distinguishes between "User 1" that provides the file and "User 2" that receives the file. (Kessler, Col. 4, line 63 to Col. 5, line 28.) The public and secret keys associated with "User 2" are PK2 and SK2, according to Kessler. The other entity in Kessler is a server that provides a proprietary client software package (Col. 4, lines 42 to 46), which as discussed more completely below; acts as an intermediary between users for transfer of public keys (Col. 5, lines 34, 35), and for accessing files (Col. 5, lines 16 to 27); and potentially for encrypting key TK (Col. 5, lines 37, 38).

To suggest the method of Claim 1, the rejection must rely upon processes performed on the client computer of "User 1" as this is the system that provides the file, according to Kessler. Reliance upon operations performed on any other entity, as was done in the rejection, changes the principles of operation of Kessler and so is inappropriate according to the MPEP. See MPEP §2143.01 VI., as quoted below.

The rejection relies upon components on two different systems which teaches away from the common entity of the claims

As quoted above, the rejection stated "*the first decryption key, i.e. SK2.*" Therefore, the rejection correlates the first cryptographic key of Claim 1 to be key SK2. Kessler taught that key SK2 is not available on the client computer of "User 1." (Kessler, Col. 4, lines 46, 47.)

Kessler also taught that key TK is generated on the client computer of "User 1." (Kessler, Col. 5, line 29; Col. 7, lines 56-57; Col 9, lines 32-33; and Col. 11, lines 50-51.). Kessler further taught that preferably encrypting operations on key TK were performed on the client computer of "User 1."

Therefore, the rejection relies upon a secret key SK2, which is only available on the client computer of "User 2," as suggesting the first cryptographic key of Claim 1 and a key TK that is generated and encrypted only on the client computer of "User 1" as suggesting the second cryptographic key of Claim 1.

However, in Claim 1, both the first and second cryptographic keys are on the application program provider, while the rejection relies on elements on two different entities.

According to the rejection, "User 1" is equivalent to the application program provider based on the encrypting of key TK that is done on the client computer of "User 1." Since key SK2 is not available to "User 1," the rejection requires a modification to Kessler to make key SK2 available to "User 1,"

because otherwise what the rejection identifies as the first and second cryptographic keys are not available on a common entity as recited in Claim 1. Making secret key SK2 of "User 2" available to "User 1" destroys the intent of Kessler that key SK2 is a secret key available only to "User 2." Thus, the characterization of Kessler in the rejection requires a change in the principles of operation of Kessler so that key SK2 is available to "User 1."

According to the MPEP:

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.

MPEP §2143.01 VI., 8th Ed., Rev. 6, pg. 2100-141, (Sept. 2007). This alone is sufficient to overcome the rejection, because the teachings of Kessler are not sufficient to support the rejection and so a *prima facie* obviousness rejection has not been made.

The rejection relies upon processes on multiple different systems which teaches away from the common entity of the claims

In addition to extracting elements available on different entities and using those elements in locations different from that required by Kessler, the rejection also confuses processes performed on different entities. Specifically, the rejection confuses processes performed on the client computer of "User 1," processes performed on the client computer of "User 2," and the processes, which are not disclosed by Kessler, used in the generation of the proprietary software package provided by Kessler, which are performed on some other entity that is different from the client computers of "User 1" and "User 2."

The rejection attempts to hide the confusion by making assertions that are incorrect and not based on any teaching or suggestion in Kessler.

For example, the rejection, as quoted above, stated "*This implies that when the proprietary software was created, not was the source code to the decryption algorithm received.*" While this may be correct, it is not relevant unless there is some suggestion that, as recited in Claim 1, a key decryption program was created on an application program provider. There is no teaching or suggestion cited that the source code was created on the client computer of "User 1" that has been identified in the rejection as the application program provider.

Specifically, first Kessler taught:

Each client computer included in the peer-to-peer file sharing environment must register with the application server 200. As part of the registration process, the client computer receives proprietary client software for facilitating the file sharing process. The client software includes a unique secret key, SK, and a unique public key, PK, for each registered client computer. The secret key and the public key are both stored in a database maintained by the server and at the registered client computer. The PK of a receiving computer is preferably provided by the server to a sending computer within the peer-to-peer file sharing environment when data is to be transferred from the sending computer to the receiving computer. The data to be transferred is encrypted using the public key, PK, of the receiving computer. Once the receiving computer receives the encrypted data, the secret key, SK, of the receiving computer is used to decrypt the data. The secret key and the public key are embedded within the client software such that neither are accessible by the user of the client computer. Such secure measures prevents unauthorized use of the secret key to decrypt the transferred data.

Kessler, Col. 4, lines 42 to 62.

Kessler further explained:

As previously discussed, each registered client computer receives proprietary client software. . . . To provide additional security, the proprietary client software also includes obfuscation and de-obfuscation algorithms. . . . The obfuscation algorithm, the de-obfuscation algorithm, the secret key, and the public key are themselves obfuscated and/or encrypted within the proprietary client software used by the client computer. In this manner, users are unable to perform de-obfuscating and decrypting that might lead to unauthorized file sharing.

Kessler, Col. 8, lines 50 to 67.

Thus, Kessler taught that each client computer **received** (not created) the same proprietary client software for facilitating the file sharing process and this software included a unique public key PK and a unique secret key SK for the client as well as encryption and decryption algorithms, and obfuscation and deobfuscation algorithms. No teaching or suggestion has been cited that any decryption program was created on the client computer, and in fact, Kessler taught that it was created elsewhere and supplied to the client computer as part of a registration process with a server.

Claim 1 recites that the decryption program is created on the application program provider. Thus, the fact that the software code for the package was received by some entity that is not what the rejection identified as the application program provider has no relevance to any process on a client computer of Kessler.

The first key is identified as secret key SK2 of "User 2" in the rejection. Thus, to teach or suggest the element of Claim 1, the rejection must cite to creating a key decryption program for key SK2 on the client computer of "User 1," which the rejection identifies as the application program provider, as noted above." The only key decryption program is described as being in the proprietary client software package of Kessler and that package is not taught by Kessler as being either created on, or sent by the client computer of "User 1."

However, Claim 1 recites that a single entity does both the creating and the sending.

The rejection relies upon Kessler as suggesting the sending, i.e., "Kessler discloses sending said key decryption program." (Final Office Action, Pg. 8, November 26, 2007.) Thus, the rejection reduces the explicit claim limitation to a gist that something does the sending. The advisory action reinforces this correct interpretation by stating "The embedded decryption algorithms . . . being considered the instruction stream . . ." Advisory Action dated February 8, 2008, bottom of page 5. Again, this confuses the peer-to-peer file sharing with the action of providing the proprietary client software. The key TK of Kessler is used in the peer-to-peer file sharing, but the proprietary client software that is considered the instruction stream of Claim 1 is not sent anywhere by the client system of "User 1."

Specifically, Claim 1 recites that the obfuscated key description program is sent from the application program provider. In contrast, the decryption program in the client software package of Kessler was not provided or sent by the client computer of "User 1," but rather was received in the registration process from a server and in fact the decryption program was not used by "User 1" in the process of providing a file. Therefore, Kessler, taken as a whole, does not support the interpretation in the above quoted rejection.

Kessler taught away from obfuscating key TK

Claim 1 also recites that not only is the second cryptographic key encrypted but also the encrypted second cryptographic key is scrambled into the instruction stream that is key decryption program for the first key. As quoted above, the rejection also identifies key TK of Kessler as being the second key of Claim 1.

Again, as cited above, key TK is created and encrypted on the client computer of "User 1" in a peer-to-peer file transfer. Thus, the rejection again confuses the supplying of the proprietary client software package with the peer-to-peer file transfer and extracts pieces from these two different processes of Kessler to arrive at Claim 1.

Moreover, Kessler expressly taught away from obfuscating key TK when that key was transferred by "User 1." Key TK is only taught as being encrypted using key PK2 when transferred. This is represented in Kessler as $E(PK2, TK)$. At Col. 9, Kessler expressly considers obfuscation and teaches that only the file being transferred is obfuscated. Kessler expressly stated " $E(PK2, TK)$ and $O2(E(TK, O1(\text{musicfile.mp3})))$ are then sent to "User 2." Col. 9, lines 51 and 52. Thus, Kessler expressly considered obfuscation and taught that obfuscation of the key TK was unnecessary, because key TK was only encrypted. Therefore, Kessler, considered as a whole, taught that obfuscation of the second key was unnecessary.

The processes associated with secret key SK2 are not on the application provider as characterized by the rejection.

Finally, the secret key SK2 cited in the rejection is used only by the client computer of "User 2," and the cited processes are performed on the client computer of "User 2," which is not what the rejection considers the "application program provider." For example, when the rejection stated "*The proprietary software contains decryption algorithms used to access the encrypted data via keys SK2 and TK,*" the rejection is relying upon processes performed on the client computer of "User 2." Actions on the client computer of "User 2" are not being performed by what the rejection identified as the application program provider, the client computer of "User 1."

Thus, the rejection selectively extracts (1) a piece of Kessler that is performed on an undefined entity (the generation of the proprietary software); (2) a piece of a peer-to-peer file transfer process on the client computer of "User 1" (the generation and encryption of key TK); and (3) a key SK2 and processes performed on the client computer of "User 2." There is no rationale for the selective extraction of different pieces of different processes from Kessler and then recombining them to read on Applicant's claims. Moreover, such a recombination as noted above changes the principles of operation of Kessler and so according to the rejection the modifications cannot support a prima facie obviousness rejection. Moreover, any one of the multiple errors is sufficient alone to overcome the rejection.

Finally, with respect to Kessler, Applicant notes that in an obviousness rejection, the MPEP is unambiguous--the reference must be considered as a whole. The above consideration of the reference as a whole and the demonstration of mischaracterizations of the reference are not considering the reference alone, but instead are comparing the characterization of the reference in the rejection with what is actually shown in Kessler, when Kessler is considered as a whole. While KSR changed the standard for combining references, it does not permit the wholesale lack of consideration of the reference as a whole.

The MPEP specifically provides that the level of analysis of Kessler in the rejection is not acceptable. Kessler has been mischaracterized and not considered as a whole as required in an obviousness determination. Moreover, it has been demonstrated that the rejection requires changes to the principles of operation of Kessler and so the rejection is not a prima facie rejection.

These facts alone are sufficient to overcome the obviousness rejections because the additional information

relied upon in the three secondary references does not fix the basic shortcomings of Kessler. Further, an examination of the various combinations demonstrates that the combinations are not well founded and that the references are also mischaracterized, when considered as a whole.

There is no basis for the combination of Kessler and Okada.

Despite the explicit teaching of Kessler, the rejection used Applicant's claim language as a road map to modify Kessler. The rejection exacts a piece from Okada which is controlling access to data on a disk drive system and which has nothing to do with peer-to-peer file transfer, as in Kessler. The rejection stated "However, Okada discloses using a first cryptographic key, i.e. session key, to both encrypt and decrypt a second key, i.e. content key (col 9, lines 21-26 and col 10, lines 39-53)." (Final Rejection, Page 8, second full paragraph.)

The rejection mischaracterizes the teaching of Okada.

This statement in the rejection mischaracterizes the teachings of Okada. Okada first taught:

The contents key to be stored in the storage section 150 is, as hereinafter described, encrypted with a host ID (predetermined access apparatus identification information), the session key and the first drive ID of the drive 100.

Okada, Col. 9, lines 22 to 26.

Thus, Okada stated that the encryption was with a host ID, the session key and the first drive ID. Next Okada did not teach that the content key was decrypted using the session key as stated in the rejection, but rather:

The decryption section 271 decrypts the encrypted contents key read out from the storage section 150 (key file 151) of the drive 100 using the drive key (equivalent) from the encryption section 242. (Emphasis Added.)

Okada, Col. 10, lines 39 to 42.

The "drive key (equivalent)" is defined as:

The encryption section 242 encrypts the first drive ID stored in the RAM 210 using session keys stored in the RAM 210 to generate a drive key (equivalent), and outputs the drive key (equivalent). The encryption section 243 further encrypts the encryption result by the encryption section 241 using the drive key (equivalent) from the encryption section 242

Okada, Col. 9, lines 44 to 47.

Thus, contrary to the statement in the rejection, the cited section of Okada directly contradicts the conclusion. The rejection reduces the teaching of Okada to a gist "using a session key" and ignores for example that the drive key uses "session keys," i.e., more than one session key. The rejection also mischaracterizes how Okada stated the encryption and decryption was actually done for the contents key.

The rejection selectively extracted only a piece of what was described as being used in the encryption, a "session key" and then mischaracterized how Okada taught the process of encryption and decryption. Thus, it has been demonstrated that the rejection failed to consider the reference as a whole. Therefore, the rejection is improper according to the requirements of the MPEP.

The combination changes the principle of operation of Kessler and so is not appropriate.

The rationale for the combination is an unsubstantiated claim of a performance advantage. Many modifications to

Kessler could produce a performance advantage, but if a modification reduced the security level taught by Kessler, the modification changes the principles of operation and so would not be appropriate according to the above quoted section of the MPEP. The rejection does not even consider the potential reduction in security and relies only on performance as a rationale for modifying Kessler. However, such performance requires changes to the principles of both Okada and Kessler and so is inappropriate.

Okada, taken as a whole, taught the level of security was sufficient when a common storage area on disk drive was used, i.e., the contents key was not transferred. Specifically, the rejection ignores that the encrypted contents key is stored in a storage section 150 of a disk drive by Okada (See Col. 9, lines 22, 23) and read from that same storage section on the same disk drive (See Col. 10, line 40). The encrypted contents key of Okada is not incorporated into any instruction stream and is not sent anywhere. Rather, the key is taught by Okada as being maintained on a disk drive. Therefore, Okada, taken as a whole, taught that this level of security was suitable when both entities accessed the same disk drive and so taught away from use of this level of security for any transfer over a network.

Without even considering the security implications, the rejection throws away the public-private key pairs of Kessler and uses the session key of Okada in their place, while ignoring the environment in which Okada taught such use was appropriate. This completely changes the principles of operation of Kessler as a session key would not be suitable for use in the "proprietary software package provided to each client," by Kessler as the same session key would not help to differentiate between users as in the unique key pairs used by Kessler.

As noted above, Kessler expressly taught that each client had a unique pair of keys. Giving each client a same key, a session key, instead of the unique pair of keys completely changes the principles of Kessler.

The modification would in fact reduce the security level by using a common session key so any party that had the session key could decrypt the file rather than a single user as in Kessler. In addition, contrary to the statements in the rejection, Okada taught that more than a session key was used. Therefore, the proposed modification changes the principles of operation of both Kessler and Okada and was directly taught away from by Okada.

According to the MPEP,

If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.

MPEP §2143.01 VI., 8th Ed., Rev. 6, pg. 2100-141, (Sept. 2007). Thus, not only was Okada not considered as a whole, but also the modifications necessary to make the reference applicable to Applicant's claims change the principles of operation of Kessler and so according to the MPEP a *prima facie* obviousness rejection has not been made.

The Advisory Action statements rejecting this argument contradict the rejection.

As noted above, the rejection cited the second cryptographic key of Claim 1 as being key TK of Kessler. Kessler taught that on the client computer of "User 1," key TK was encrypted using the public key PK2 of "User 2." The computer system of "User 2" would decrypt the encrypted key TK using secret key SK2 of "User 2." The rejection characterized

the public key-private key combination used in the encryption and decryption of key TK as asymmetric and the rejection suggested replacing these with a session key of Okada, i.e.,

It is noted that in Kessler's invention, a public key PK2 is used to encrypt the second cryptographic key, i.e. TK, while a secret key SK2 is used to decrypt the second cryptographic key. In the invention recited in claim 1, a first cryptographic key is used to both create the encrypted second key and to decrypt the second key, i.e. perform said decryption algorithm for said first cryptographic key. However, Okada discloses using a first cryptographic key, i.e. session key, to both encrypt and decrypt a second key, i.e. content key (col 9, lines 21-26 and col 10, lines 39-53). Note that the content key disclosed by Okada is equivalent to the track key (TK) disclosed by Kessler.

At the time applicant's invention was made, it would have been obvious to one of ordinary skill in the art to modify Kessler's invention such that rather than use an asymmetric key system to encrypt/decrypt the second cryptographic key TK, . . .

Final Office Action, Pgs. 8, November 26, 2007.

Thus, the rejection relies upon replacing the keys used in encrypting and decrypting key TK with a different key.

However, the advisory action stated the contrary:

Applicant argues that the public key PK and secret key SK was needed to make file transfer work, thus use of other key combination (i.e. TK) would break Kessler because the public key PK and secret key SK combination is gone. The examiner respectfully disagrees. Kessler's invention actually requires use of other keys, i.e. TK, for file transfer. Figure 2 shows that each client contains a respective public and secret key pair (i.e. PK1, SK1, PK2, and SK2). One skilled should understand that in public key cryptography, a file that has been encrypted using a public key can only be decrypted using the corresponding secret/private key. In other words PK and SK referred to by Kessler are public/private key pairs. The clients, having these keys, are disclosed by Kessler as also using TK for file transfer (Fig 3; col 5, lines 29-42; and col 6, lines 52-65). Since use of TK is required to make Kessler's invention work, it's use would not remove PK and SK from his invention and break the invention. In

Kessler's invention, the public/private key pair is used to transfer TK from one client to the other while TK itself is used to encrypt/decrypt the file being transferred, i.e. musicfile.mp3.

Advisory Action dated February 8, 2008, bottom of page 6.

The rejection proposed replacing the public/private key pair. The advisory action makes it clear that this pair is essential and so in fact agrees with Applicant. Applicant did not and does not argue about key TK, but rather that the proposed modification to Kessler with respect to encryption of key TK is inappropriate. The above comments fail to address the substance of the remarks and instead argue about key TK, which was not even at issue in this portion of the rejection. Substitution of the session key of Okada into the encryption of key TK in Kessler removes the public key and the secret key, and so the advisory action response makes no sense.

Moreover, Kessler expressly considered public key encryption and symmetric key encryption and stated:

. . . two different encryption technologies are used to provide this secure peer-to-peer filing sharing environment. The first uses public key encryption to encrypt and decrypt the track key, TK. The second uses symmetric encryption to encrypt and decrypt the music file, musicfile.mp3. Symmetric encryption uses the same key, in this case TK, to both encrypt and decrypt.

Kessler, Col. 6, lines 59 to 65.

Thus, Kessler was familiar with both public key encryption and symmetric encryption and taught which encryption was best in the different circumstances. This is further evidence that the modification in the rejection changes the principles of operation of Kessler.

In particular, the rejection proposes to change the public key encryption of Kessler to a symmetric key encryption based upon a purported performance advantage. Kessler taught that two different encryption technologies were used. Thus, to

change the two encryption technologies to the same technology, symmetric encryption, is further direct evidence that the combination changes the principles of operation of Kessler and so is inappropriate according to the MPEP.

The combination with Shen Orr is based on mischaracterizations and so violates the as a whole requirement.

The mischaracterization of the prior art continues. In the combination of Kessler, Okada, and Shen Orr, the rejection cited page 16, line 31 to page 17, line 3; page 21, lines 1 to 2; and page 24, lines 17 to 19 of Shen Orr. These portions stated:

According to still another preferred embodiment of the present invention, there is provided an improvement for a method for protecting software code, the method protecting the software code by obfuscating at least a portion of the software code, the improvement comprising providing a plurality of magazines, each magazine comprising at least one instruction for obfuscating at least the portion of the software code, such that a plurality of versions of the software code is produced according to the plurality of magazines

The compiler preferably, additionally or alternatively, obfuscates the code of the descrambler/player. Alternatively or additionally, the descrambler/player may optionally be protected with conventional encryption mechanisms, for example in order to protect the descrambler/player for transmission to the end user, particularly for such transmission in advance of the protected content

Each magazine 130 preferably includes some type of obfuscation, such that the final compiled software is preferably obscured in some way for protection of the software from unauthorized execution, as described in greater detail below.

Again, the MPEP requires that Shen Orr be considered as a whole. The rejection had not cited any teaching or suggestion in Shen Orr of obfuscation of a key decryption program as recited in Claim 1. The reason is that Shen Orr is concerned with obfuscating a "descrambler/player." Therefore, Shen Orr provides no support for the conclusions reached.

The rejection extracts general teachings about obfuscation and then proposes to modify the primary reference to include such a technique, even though both the primary reference and Okada taught that there was no need for such a modification. Both Kessler and Okada taught that the key relied upon in the rejection was only encrypted and not obfuscated, as noted above. The rejection has not cited any teaching in Shen Orr of such a process. Thus, the three references taken together do not support the characterization used, and the two primary references teach away from the modification.

Moreover, the rationale for the modification further demonstrates that the modification to Kessler changes the principles of operation of Kessler. The rationale stated:

One skilled would have been motivated to do so because as recognized by Shen Orr, there was a need in the art to provide variable security mechanisms (p6, lines 20-26), which would provide more security than using a single security scheme such as the one used by Kessler.

Final Office Action, top of Pg. 10.

First, the rejection again confuses the proprietary software tools provided by Kessler with the transfer of a file over the peer-to-peer network. For example, this part of the rejection stated:

At the time applicant's invention was made, it would have been obvious to one of ordinary skill in the art to further modify Kessler's invention using Shen Orr's teachings by obfuscating Kessler's proprietary client software before sending it to the client . . .

Final Office Action, bottom of Pg. 9 and top of Pg. 10.

As noted above, the keys that are relied on in the rejection are used in the peer-to-peer file transfer, and the computer system of "User 1" was cited as the application program provider of Claim 1. The computer system of "User 1" does not send the propriety client software to anything. Thus, comments concerning protection of the transfer of the proprietary software package of Kessler simply demonstrate that Kessler is being modified and redefined in ways that go against the express teaching of Kessler. While, as noted above, the requirements for an obviousness rejection have not been reduced to selectively picking and choosing among references in a way that explicitly ignores what the references as a whole teach.

Applicants respectfully note that yet another reference was cited. However, the information relied upon in the final reference does not correct the multiple defects in the rejection. Any one of the distinctions noted above is sufficient to overcome the obviousness rejection. Therefore, Applicant respectfully requests reconsideration and withdrawal of the obviousness rejection of each of Claims 1, 6, 11 and 16.

With respect to each of the claims dependent from Claims 1, 6, 11 and 16, the additional material relied upon from the secondary references, or the new reference with respect to Claims 3, 8, 13 and 18, does not correct the deficiencies of the combination of references with respect to the independent claims from which these claims depend. Therefore, each of Claims 2 to 4, 7 to 9, 11 to 14 and 17 to 19 distinguish over the combination of references for at least the same reasons as the independent claims.

In conclusion, Appellant has explained at multiple levels why the combination of references fails to render the invention as recited in Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19 obvious. Thus, the Examiner's rejection of Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19 should be reversed.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

CONCLUSION

For the reasons above, all appealed claims, i.e., Claims 1 to 4, 6 to 9, 11 to 14 and 16 to 19, are allowable. Appellant respectfully requests the Board of Patent Appeals and Interferences to reverse the Examiner's rejections under U.S.C. § 103(a) of these claims.

CLAIMS APPENDIX

1. (Previously Presented) A method for application program obfuscation, comprising:

receiving, on an application program provider, a reference to a decryption algorithm and a first cryptographic key;

creating, on said application program provider, a key decryption program comprising an instruction stream, said key decryption program configured to perform said decryption algorithm for said first cryptographic key;

applying, on said application program provider, a cryptographic process to a second cryptographic key to create an encrypted second cryptographic key wherein said cryptographic process receives said first and second cryptographic keys as inputs;

scrambling, on said application program provider, said encrypted second cryptographic key into said instruction stream using a code obfuscation method indicated by an obfuscation descriptor, said scrambling creating an obfuscated key decryption program, said obfuscation descriptor based at least in part on a target ID wherein said target ID specifies a user device for executing an obfuscated application program; and

sending, from said application program provider, said obfuscated key decryption program.

2. (Original) The method of claim 1, further comprising sending digital content protected by said second cryptographic key.

3. (Original) The method of claim 2, further comprising sending said obfuscated key decryption program together with said digital content.

4. (Original) The method of claim 1 wherein said target ID comprises a VM ID.

5. (Withdrawn) A method for application program obfuscation, comprising:

receiving an obfuscated key decryption program comprising an instruction stream configured to perform a decryption algorithm for a first cryptographic key, said obfuscated decryption program having an encrypted second cryptographic key scrambled in said instruction stream, said second cryptographic key encrypted with said first cryptographic key;

executing said program to decrypt said second cryptographic key; and

decrypting digital content using said second cryptographic key.

6. (Previously Presented) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method for application program obfuscation, the method comprising:

receiving, on an application program provider, a reference to a decryption algorithm and a first cryptographic key;

creating, on said application program provider, a key decryption program comprising an instruction stream, said key decryption program configured to perform said decryption algorithm for said first cryptographic key;

applying, on said application program provider, a cryptographic process to a second cryptographic key to

create an encrypted second cryptographic key wherein said cryptographic process receives said first and second cryptographic keys as inputs;

scrambling, on said application program provider, said encrypted second cryptographic key into said instruction stream using a code obfuscation method indicated by an obfuscation descriptor, said scrambling creating an obfuscated key decryption program, said obfuscation descriptor based at least in part on a target ID wherein said target ID specifies a user device for executing an obfuscated application program; and

sending, from said application program provider, said obfuscated key decryption program.

7. (Original) The program storage device of claim 6, said method further comprising sending digital content protected by said second cryptographic key.

8. (Original) The program storage device of claim 7, said method further comprising sending said obfuscated key decryption program together with said digital content.

9. (Original) The program storage device of claim 6 wherein said target ID comprises a VM ID.

10. (Withdrawn) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method for application program obfuscation, the method comprising:

receiving an obfuscated key decryption program comprising an instruction stream configured to perform a decryption algorithm for a first cryptographic key, said obfuscated decryption program having an encrypted second cryptographic key scrambled in said instruction stream,

said second cryptographic key encrypted with said first cryptographic key;

executing said program to decrypt said second cryptographic key; and

decrypting digital content using said second cryptographic key.

11. (Previously Presented) An apparatus for application program obfuscation, comprising:

a processor; and

a memory, coupled to said processor, having stored therein computer readable instructions wherein executing said computer readable instructions on said processor provides:

means for receiving, on an application program provider, a reference to a decryption algorithm and a first cryptographic key;

means for creating, on said said apparatus, a key decryption program comprising an instruction stream, said key decryption program configured to perform said decryption algorithm for said first cryptographic key;

means for applying, on said apparatus, a cryptographic process to a second cryptographic key to create an encrypted second cryptographic key wherein said cryptographic process receives said first and second cryptographic keys as inputs;

means for scrambling, on said apparatus, said encrypted second cryptographic key into said instruction stream using a code obfuscation method indicated by an obfuscation descriptor, said scrambling creating an obfuscated key decryption program, said obfuscation descriptor based at least in part on a target ID wherein said target ID

specifies a user device for executing an obfuscated application program; and

means for sending, from said apparatus, said obfuscated key decryption program.

12. (Original) The apparatus of claim 11, further comprising means for sending digital content protected by said second cryptographic key.

13. (Original) The apparatus of claim 12, further comprising means for sending said obfuscated key decryption program together with said digital content.

14. (Original) The apparatus of claim 11 wherein said target ID comprises a VM ID.

15. (Withdrawn) An apparatus for application program obfuscation, comprising:

means for receiving an obfuscated key decryption program comprising an instruction stream configured to perform a decryption algorithm for a first cryptographic key, said obfuscated decryption program having an encrypted second cryptographic key scrambled in said instruction stream, said second cryptographic key encrypted with said first cryptographic key;

means for executing said program to decrypt said second cryptographic key; and

means for decrypting digital content using said second cryptographic key.

16. (Previously Presented) An apparatus for application program obfuscation, comprising an application program provider comprising:

a processor; and

a memory, coupled to said processor, having stored therein computer readable instructions wherein executing said computer readable instructions on said application program provider is configured to:

receive, on an application program provider, a reference to a decryption algorithm and a first cryptographic key;

create, on said application program provider, a key decryption program comprising an instruction stream, said key decryption program configured to perform said decryption algorithm for said first cryptographic key;

apply, on said application program provider, a cryptographic process to a second cryptographic key to create an encrypted second cryptographic key wherein said cryptographic process receives said first and second cryptographic keys as inputs;

scramble, on said application program provider, said encrypted second cryptographic key into said instruction stream using a code obfuscation method indicated by an obfuscation descriptor, said scrambling creating an obfuscated key decryption program, said obfuscation descriptor based at least in part on a target ID wherein said target ID specifies a user device for executing an obfuscated application program; and

send, from said application program provider, said obfuscated key decryption program.

17. (Original) The apparatus of claim 16, said application program provider further configured to send digital content protected by said second cryptographic key.

18. (Original) The apparatus of claim 17, said application program provider further configured to send said obfuscated key decryption program together with said digital content.

19. (Original) The apparatus of claim 16 wherein said target ID comprises a VM ID.

20. (Withdrawn) An apparatus for application program obfuscation, comprising a target device configured to:

receive an obfuscated key decryption program comprising an instruction stream configured to perform a decryption algorithm for a first cryptographic key, said obfuscated decryption program having an encrypted second cryptographic key scrambled in said instruction stream, said second cryptographic key encrypted with said first cryptographic key;

execute said program to decrypt said second cryptographic key; and

decrypt digital content using said second cryptographic key.

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

Revised Appeal Brief Filed: June 25, 2008

EVIDENCE APPENDIX

None

Serial No. 10/672,184

Notice of Non-Compliant Appeal Brief: May 29, 2008

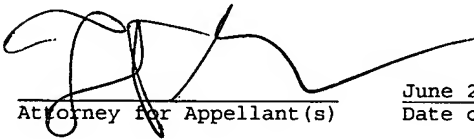
Revised Appeal Brief Filed: June 25, 2008

RELATED PROCEEDINGS APPENDIX

None

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 25, 2008.


Attorney for Appellant(s)

June 25, 2008
Date of Signature

Respectfully submitted,



Forrest Gunnison
Attorney for Appellant(s)
Reg. No. 32,899
Tel.: (831) 655-0880